Secure Multiparty Computation

Muhammad Naveed

?

Please Interrupt

Millionaire's Problem

Millionaire's Problem

Millionaire's Problem

Millionaire's Problem

Let's see who is richer

Millionaire's Problem

OK, tell me your wealth.

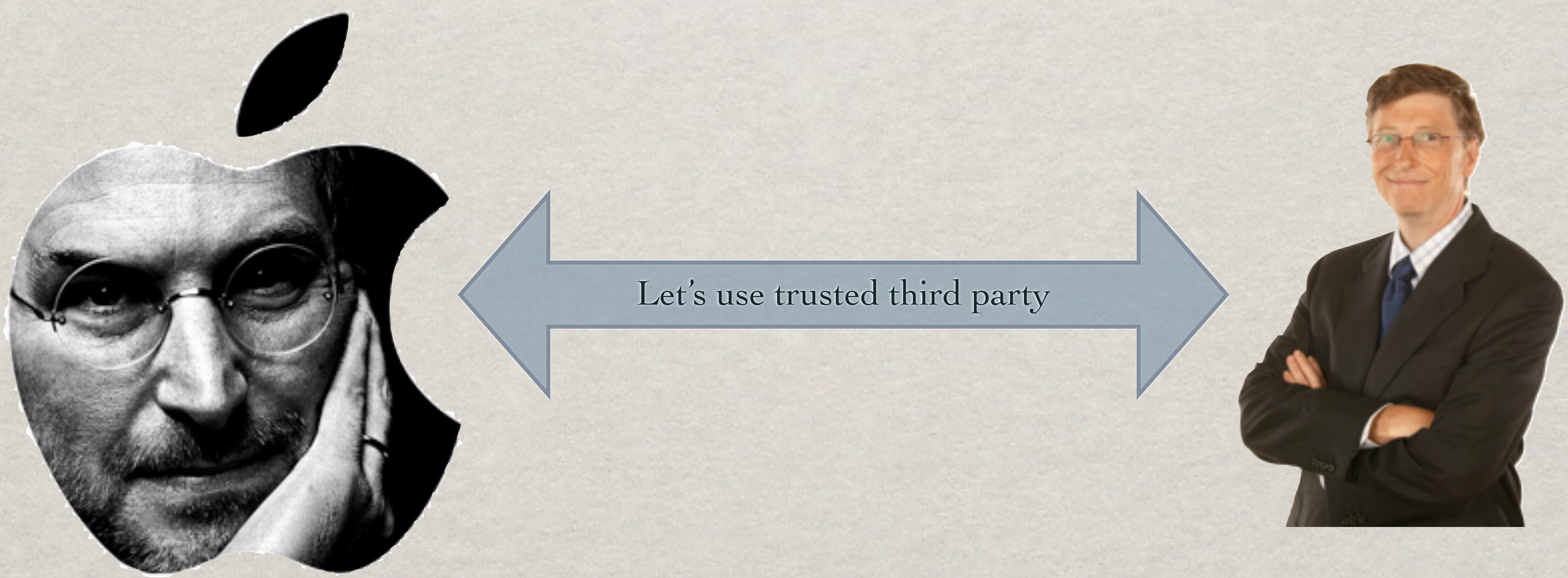
Millionaire's Problem

NO, tell me your wealth.

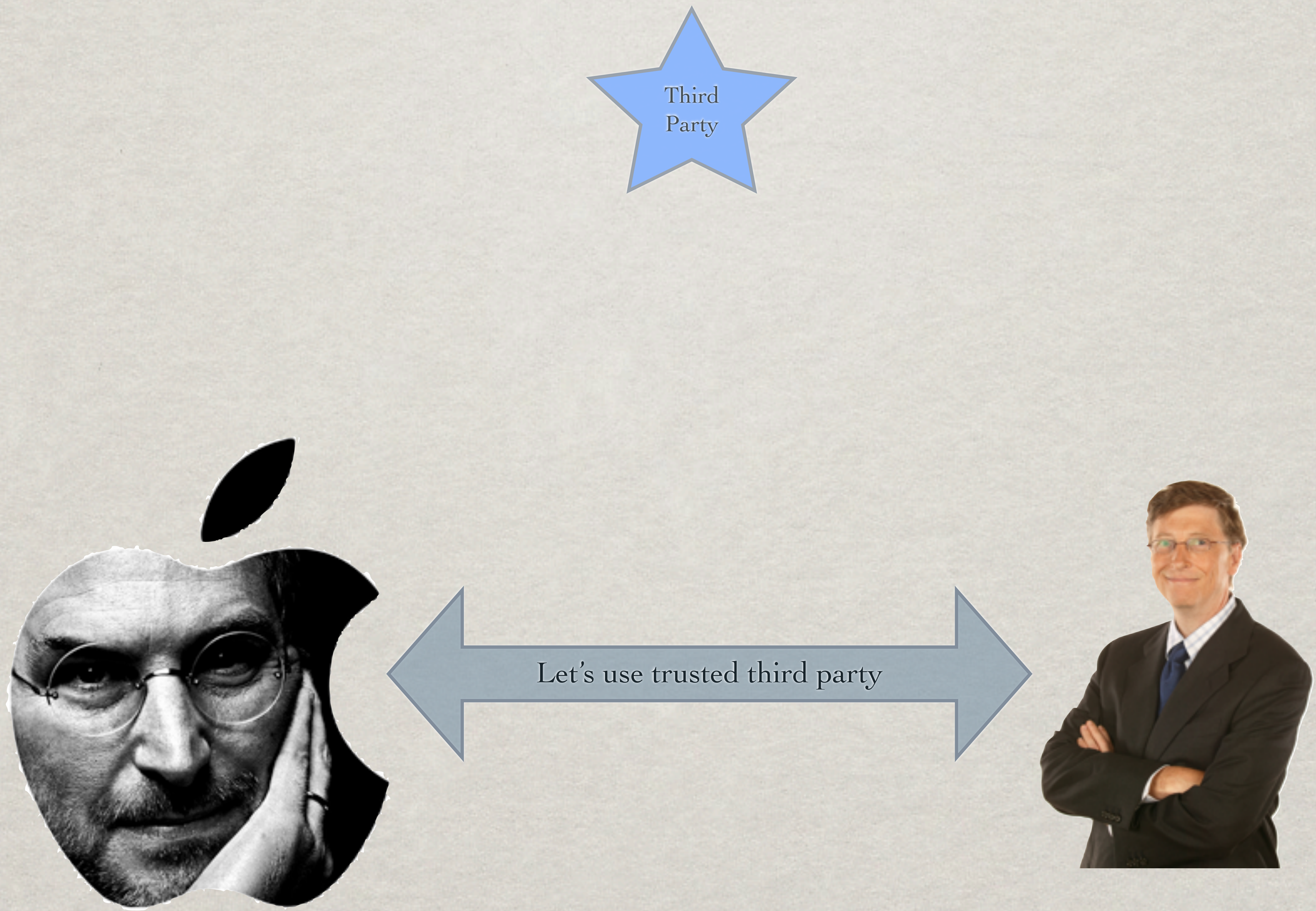
Millionaire's Problem

Trusted Third Party

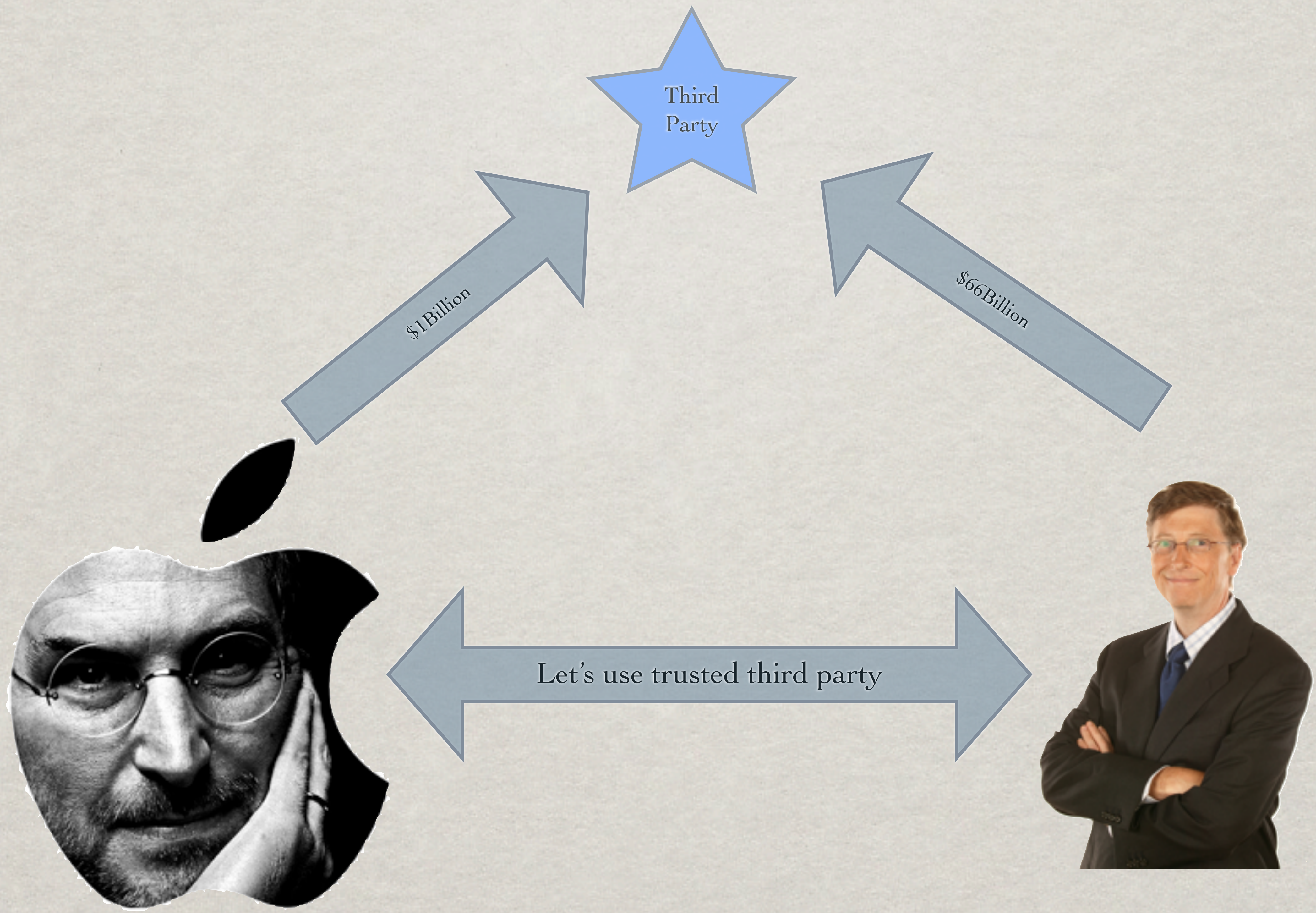
Trusted Third Party



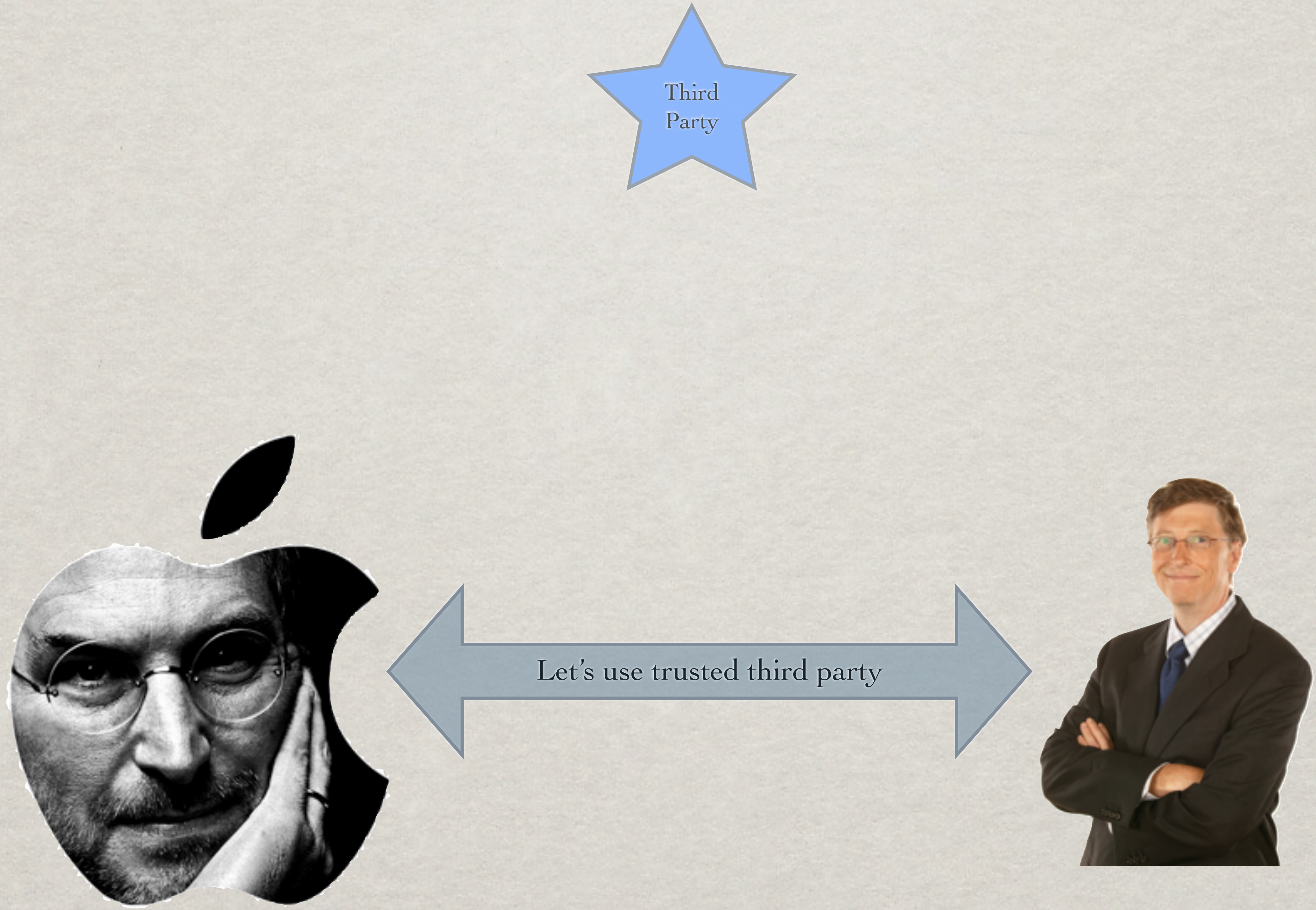
Trusted Third Party



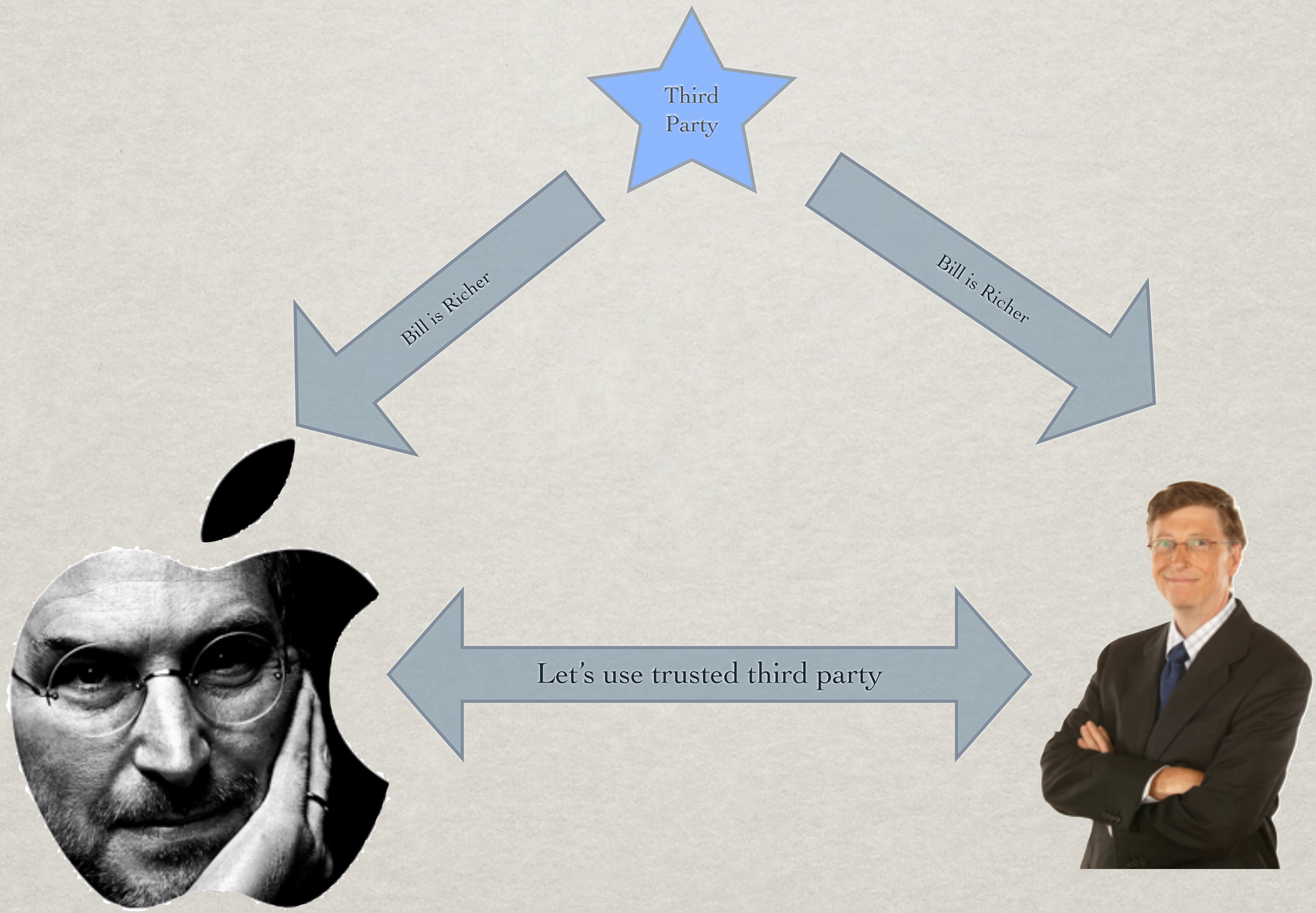
Trusted Third Party



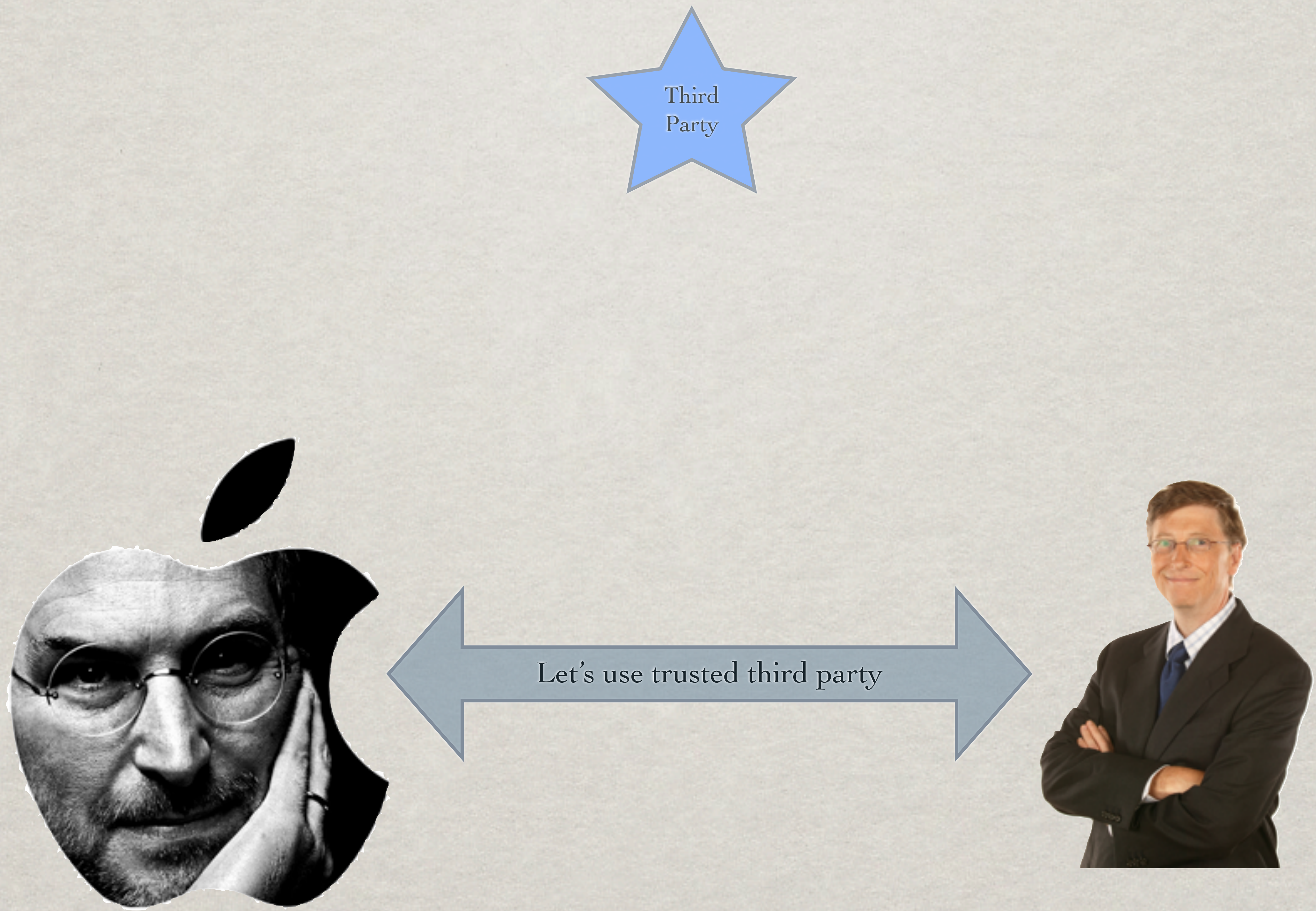
Trusted Third Party



Trusted Third Party



Trusted Third Party



Trusted Third Party

Third Party

Forbes

Let's use trusted third party

Trusted Third Party

Third Party

Forbes

2		**Bill Gates**	$67 B	57	Microsoft	United States

Let's use trusted third party

Secure Multiparty Computation

- Yao's Garbled Circuits [Yao1982]
  - solves Millionaire's Problem
  - first secure multiparty computation scheme
  - can compute any function securely
  - doesn't leak anything about inputs, other than what output leaks
  - security only in honest but curious model

Andy Yao

Secure Multiparty Computation

- Yao's Garbled Circuits [Yao1982]
  - solves Millionaire's Problem
  - first secure multiparty computation scheme
  - can compute any function securely
  - doesn't leak anything about inputs, other than what output leaks
  - security only in honest but curious model

Guys, you don't need third party.

UIUC Alumni

Andy Yao

Applications

- Auctions
- Electronic Voting
- Genomic Computation

Applications

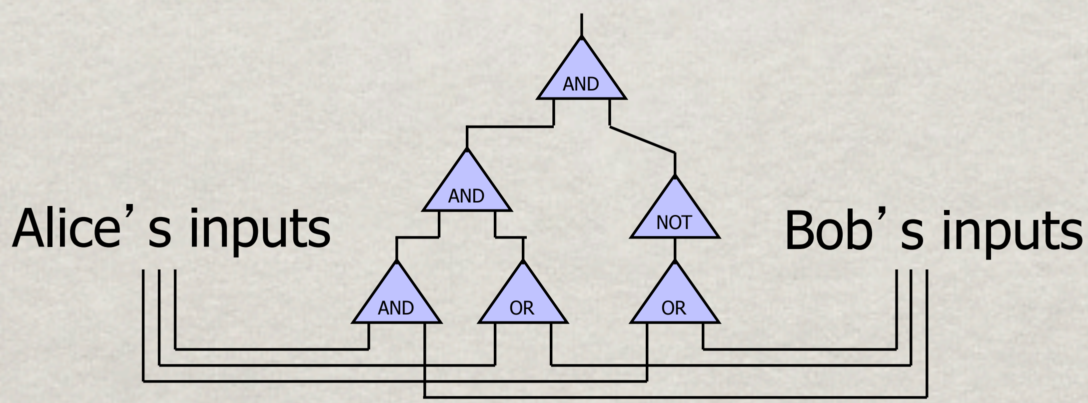
- Auctions
- Electronic Voting
- Genomic Computation
- Space Security

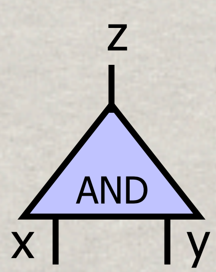
Applications

- Auctions
- Electronic Voting
- Genomic Computation
- Space Security
    - Sharing information between satellites to avoid collision but not sharing trajectories [http://sharemind.cyber.ee/]

Yao's Garbled Circuits

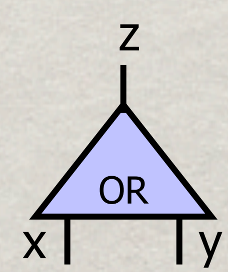
- First convert circuit into boolean circuit





Truth table:

x	y	z
0	0	0
0	1	0
1	0	0
1	1	1



Truth table:

x	y	z
0	0	0
0	1	1
1	0	1
1	1	1

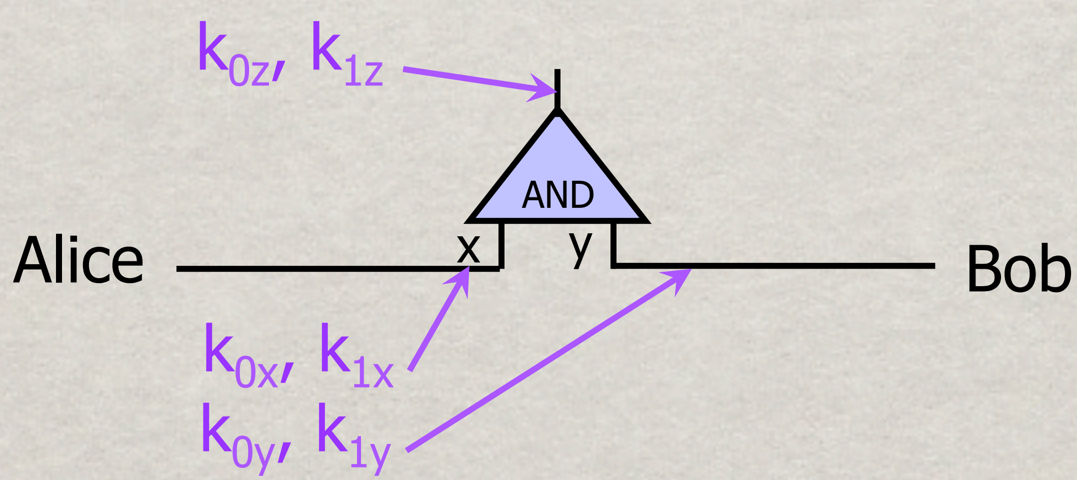
Slide adapted from Vitaly Shmatikov Slides

YAO'S PROTOCOL

- Consider a two input AND gate
  - same idea extends to larger circuits
- Alice have bit b_A and Bob with bit b_B wants to compute b_A AND b_B
- Two parties:
  - Generator generates the circuit
  - Evaluator evaluate the circuit
- Any party can generate the circuit and the other party evaluates the circuit

Garbling Input

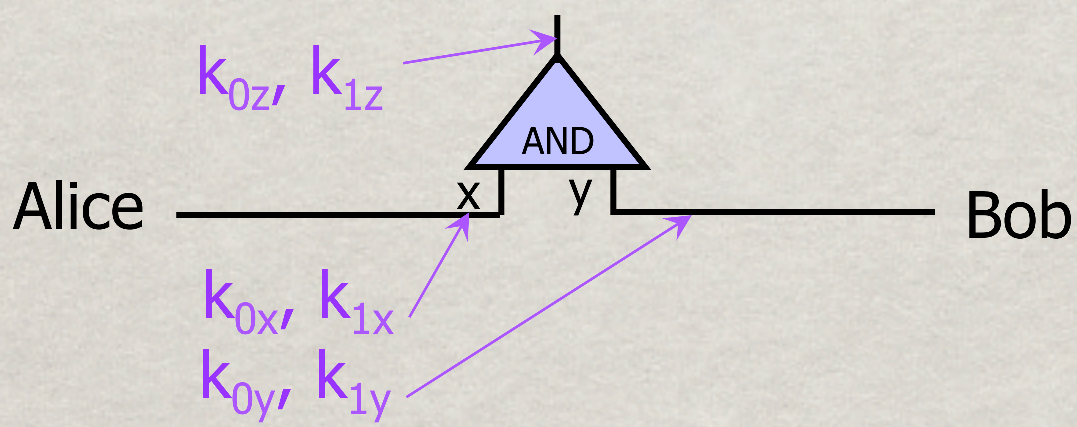
- Without loss of generality, suppose Alice generates the circuit
- Alice will pick two random keys for all wires of the gate



Slide adapted from Vitaly Shmatikov Slides

Garbling the circuit

- Alice encrypts each row of the truth table with encrypting the output wire key with the corresponding input wire keys



Original truth table:

x	y	z
0	0	0
0	1	0
1	0	0
1	1	1

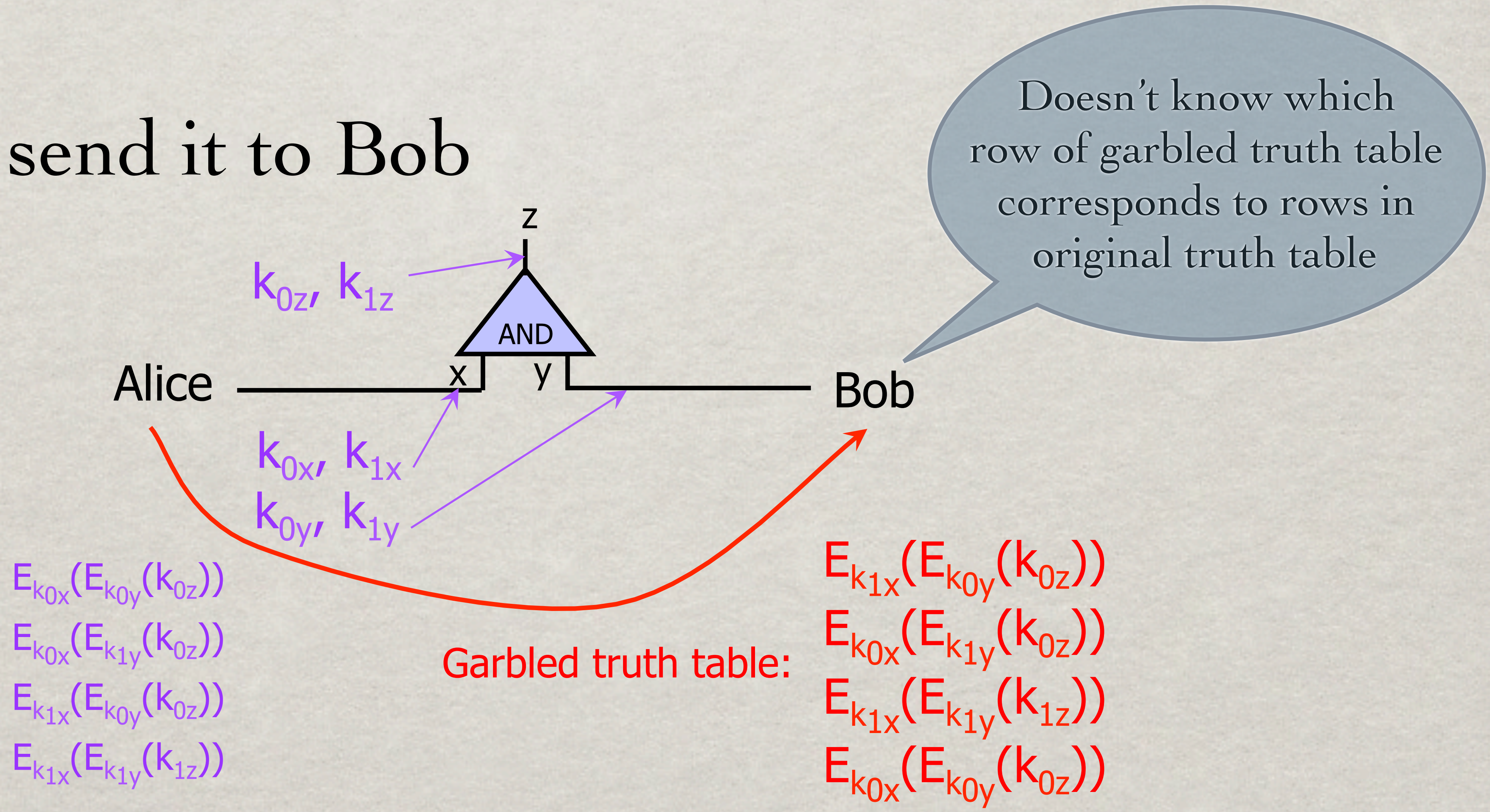
Encrypted truth table:

$E_{k0x}(E_{k0y}(k_{0z}))$
$E_{k0x}(E_{k1y}(k_{0z}))$
$E_{k1x}(E_{k0y}(k_{0z}))$
$E_{k1x}(E_{k1y}(k_{1z}))$

Slide adapted from Vitaly Shmatikov Slides

Send Garbled Circuit to Bob

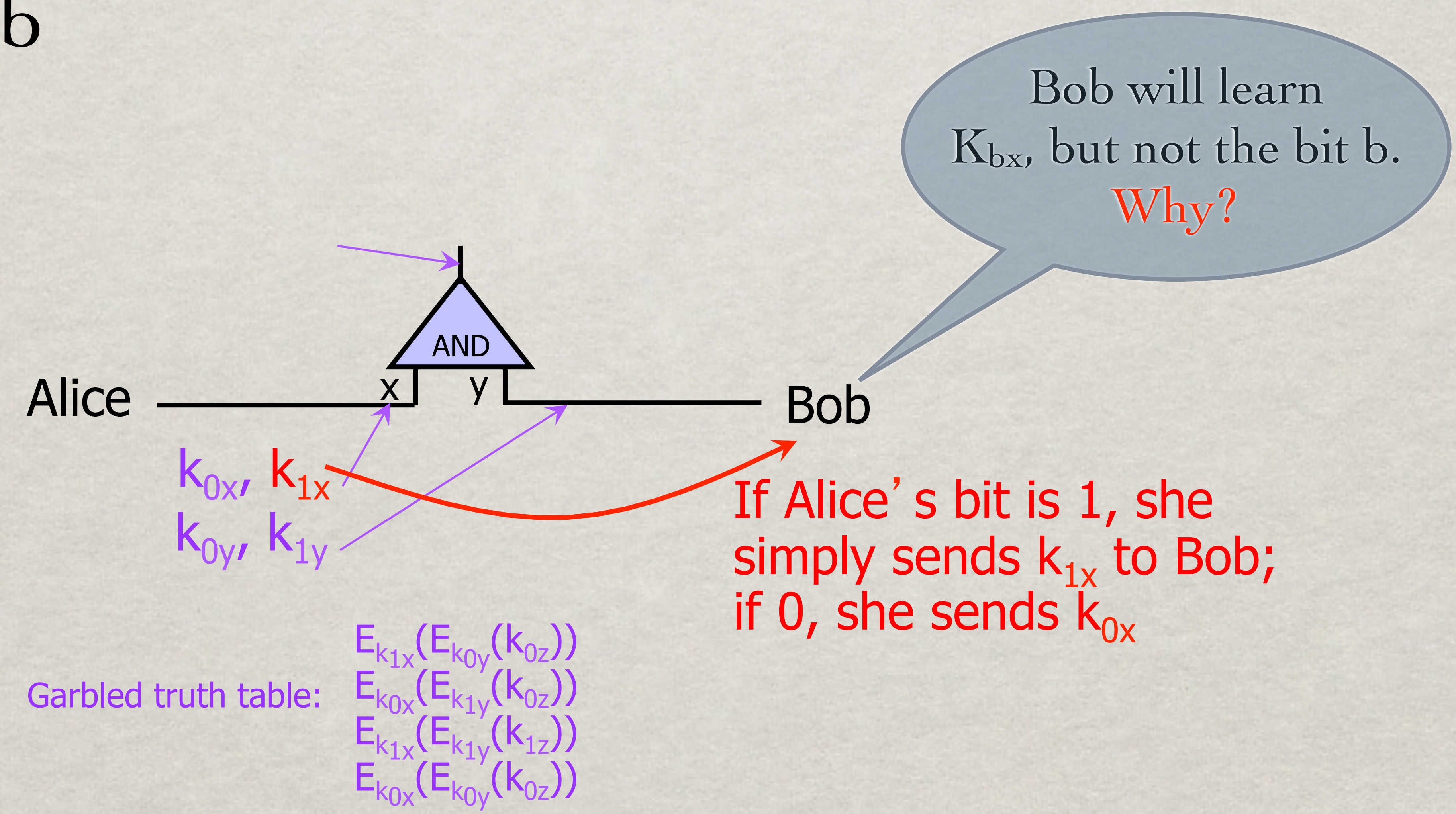
- Alice randomly permute the garbled truth table
- And send it to Bob



Slide adapted from Vitaly Shmatikov Slides

Alice send Its Keys

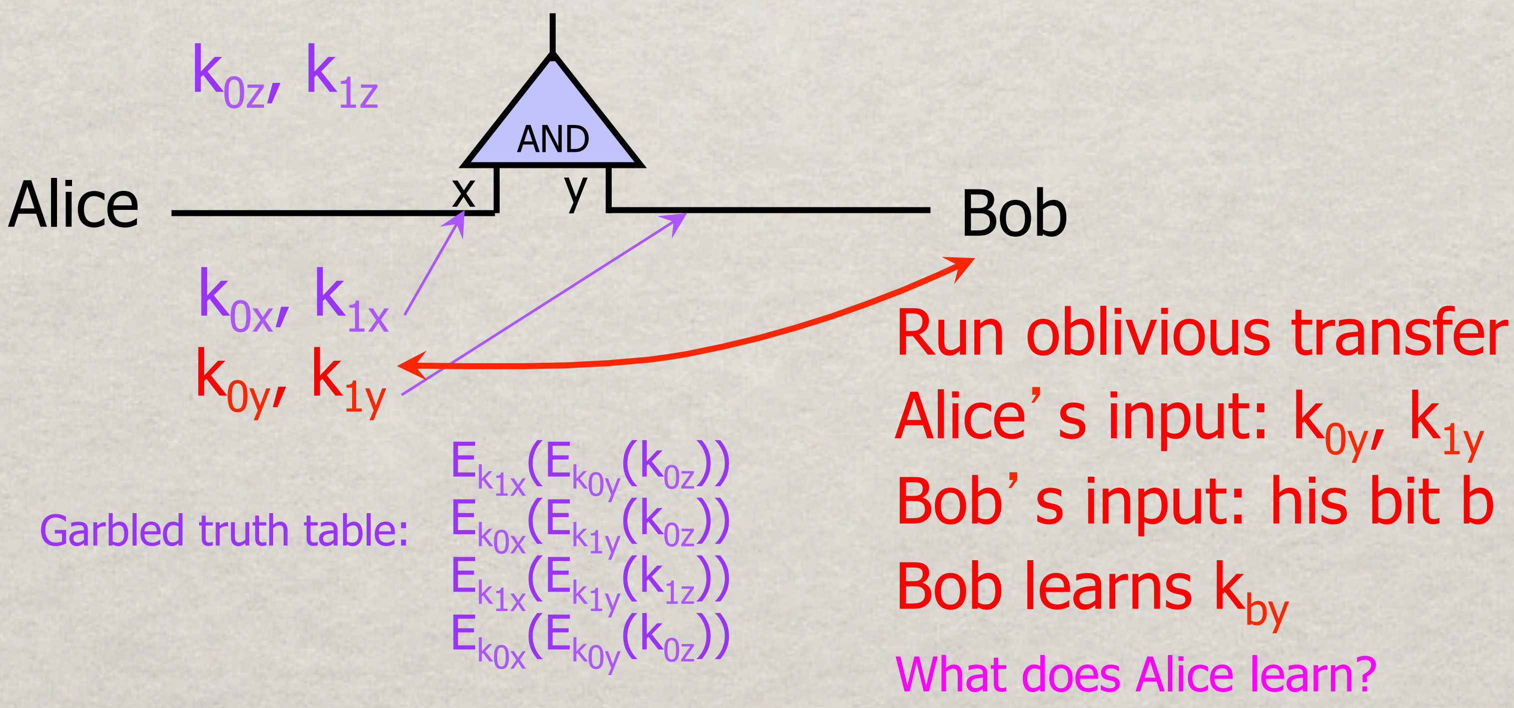
- Alice send keys corresponding to its inputs to Bob



Slide adapted from Vitaly Shmatikov Slides

Bob Get His Keys Using OT

- OT stands for oblivious transfer. Suppose,
  - 1st party has k_0 and k_1
  - 2nd party input is a bit b = 0 or 1 and wants to learn k_b
  - Using OT, second party will learn k_b, while first party will not learn b

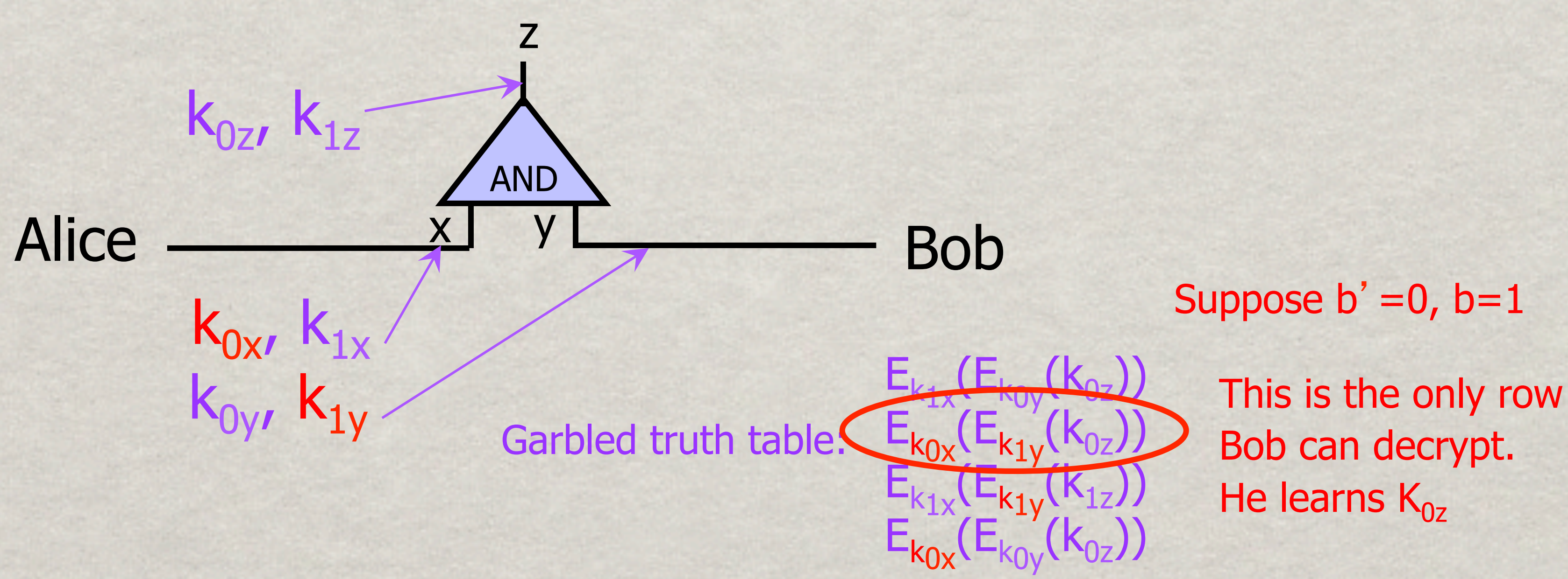


Slide adapted from Vitaly Shmatikov Slides

Evaluate Garbled Gate

- Using the two keys, bob will be able to decrypt only one entry in the truth table and will get <u>output wire key</u>

- Bob does not learn if the output wire key corresponds to 0 or 1



Slide adapted from Vitaly Shmatikov Slides

Evaluating Entire Circuit

- In the same way, Bob evaluates the entire garbled circuit
  - For each wire, Bob learns one key
  - But Bob doesn't know whether the key corresponds to 0 or 1
    - i.e. Bob doesn't know intermediate values
- Bob tells Alice the key for the final output
  - She tells him whether it corresponds to 0 or 1
  - Bob will not tell Alice the intermediate values

Slide adapted from Vitaly Shmatikov Slides

Improvements

- Yao's garbled circuit was proposed as a theoretical construction
- Real implementation is memory intensive
- Many improvements to make it more efficient and scalable
  - Garbling XOR gates for free
  - Pipelining

Reading Paper

- Yan Huang et. al. Faster Secure Two-Party Computation Using Garbled Circuits, Usenix Security 2011
- Circuit Level Optimization
  - minimize bid-width
  - exploit free XOR garbling, convert as much gates to XOR as possible
  - MultiInput/MultiOutput gates
- Program Level
  - exploit local computation

Reading Paper

- Yan Huang et. al. Faster Secure Two-Party Computation Using Garbled Circuits, Usenix Security 2011
- Circuit Level Optimization

	Hamming Distance (900 bits)		Levenshtein Distance		AES	
	Online Time	Overall Time	Overall Time†	Overall Time‡	Online Time	Overall Time
Best Previous	0.310 s [26]	213 s [26]	92.4 s	534 s	0.4 s [11]	3.3 s [11]
Our Results	0.019 s	0.051 s	4.1 s	18.4 s	0.008 s	0.2 s
Speedup	16.3	4176	22.5	29	50	16.5

Table 1: Performance comparisons for several privacy-preserving applications.

† Inputs are 100-character strings over an 8-bit alphabet. The best previous protocol is the circuit-based protocol of [16].

‡ Inputs are 200-character strings over an 8-bit alphabet. The best previous protocol is the main protocol of [16].

  - MultiInput/MultiOutput gates
- Program Level
  - exploit local computation

Interesting Problems

- SMC guarantees that nothing will be leaked about the inputs, other than the leakage from output of computation

- e.g. Alice has 3 and Bob has 5 and they want to compute SUM(3, 5) = 8

  - Alice's learns Bob's input and Bob's learns Alice's input

  - It's still perfectly secure SMC

Conclusion

- Yao's garbled circuits enable computation of any function without revealing inputs
- A constant round protocol
- Secure only against honest but curious adversaries
- State of the art SMC techniques are practically useful
- Other solutions for SMC